

**In the United States Patent & Trademark Office
Patent Application under 37 CFR 1.53(b)**

Inventor(s) Adams, et al.
Serial No.: New Application
Filed: -
For: Flexible Matrix Fabric Design Framework for Multiple Requestors and
Targets in System-on-Chip Designs
Docket No.: 63479.0116

Inventors: Lyle E. Adams
Michael Ou

Palmchip Corporation

Prepared by:

Karen S. Wright
Matthew J. Booth
Booth & Wright, L.L.P.
PO Box 50010
Austin TX 78763
Tel: (512) 474-8488
Fax: (512) 474-7996
<http://www.boothlaw.com/>

Customer No.: 23309
Deposit Account No.: 11-0851

Cross Reference To Related Applications

[01] This application claims the benefits of the earlier filed US Provisional Application Serial No. 60/421,702, filed 28 October 2002 (28.10.2002), which is incorporated by reference for all purposes into this specification.

Background Of The Invention

Field Of The Invention

[02] The present invention relates to developing system-on-chip (SOC) designs. More specifically, the present invention provides a design framework that provides designers with the flexibility to easily add multiple requestors and targets into an SOC design, thereby increasing the bandwidth and throughput of the system, without changing the architecture of the system.

Description Of The Related Art

[03] Demand for memory bandwidth is constantly increasing as applications become more complex and grow more data hungry. Faster and more advanced processors are being used to run such applications, which results in the processor requiring more system memory bandwidth for data accesses and cache lines fills. In addition, peripheral interface standards are all constantly evolving to allow for more data throughput. For example, 10/100 Ethernet with transfer rates of 10Mbits per second and 100Mbits per second of data is being replaced with the significantly faster Gigabit Ethernet and even 10 Gigabit Ethernet. The USB 1.1 interface, which has a maximum bandwidth of 12 Mbits of data per second, is being replaced by USB 2.0, which has increased bandwidth to 480Mbits per second now.

1 [04] On a separate front, design and development time for new systems is continually
2 shrinking as time-to-market demands force shortening of chip design schedules. This
3 results in conflicting design constraints, where designers must balance the need to
4 increase memory bandwidth in system designs with the constraints of shorter design
5 and development time and less complexity of design for simpler verification. Current
6 SOC designs that have architectures designed to increase memory bandwidth usually
7 are highly complex and require significantly more verification time than prior, standard-
8 bandwidth designs. In addition, these complex, high-memory-bandwidth designs lack
9 flexibility when changes need to be made to the system architecture.

10 [05] Accordingly, a design framework and approach are required that enable SOC
11 designers to efficiently develop complex, increased-bandwidth SOC designs that are
12 flexibly upgradeable, capable of efficient verification, and marketable after a reasonably
13 short development time. Ideally, such a framework would support a wide range of
14 designs and design complexity, from single target/single requestor to multiple
15 target/multiple requestor designs. It would support both original design efforts and
16 upgrades. It would enable designers to increase memory bandwidth of SOC's in
17 development by adding additional memory targets and allow additional requestors to be
18 added without affecting the design of the individual targets and/or requestors. It would
19 support multi-port devices that may be both targets and requestors. It would support
20 different bus protocols between and among the targets and requestors. It would enable
21 flexible system upgrades and modification. And finally, it would provide support for
22 arbitrary pipelining, rendering it usable for both small and large chip designs.

- 1 [06] The matrix fabric framework of the present invention is such a design framework
- 2 and approach.

Summary Of The Invention

The present invention is a System-on-Chip (SOC) interconnection apparatus and system, wherein one or more requestors and one or more addressable targets are interconnected by an internal switching fabric on a single semiconductor integrated circuit. Each target has a unique address space and may be resident (i.e., on-chip) memory, a memory controller for resident or off-chip memory, an addressable bridge to a device, an addressable bridge to a system or subsystem, or any combination thereof. Independently accessible ports on multi-port devices may also be individual targets, and some devices, such as a PCI bridge, may function both as a requestor and a target. The present invention supports targets with internal arbitration, and those without. Targets and requestors are connected to the internal switching fabric of the present invention using target connection ports and requestor connection ports.

The internal switching fabric of the present invention routes signals between requestors and targets using one or more decoder/router elements. Each decoder/router element receives a request from a requestor, determines which target is the designated target using an internal system memory map, and routes the request to the designated target. The internal system memory map used in an individual decoder/router element may include unique address space information for all of the targets in a system, or less than all of the targets in a system. A single decoder/router element may route requests to all of the targets in a system, or fewer than all of the targets in a system.

The internal switching fabric may also include independent arbiters dedicated to targets that do not have internal arbitration. Finally, the signals routed between the

1 decoder/routers and the targets by the interconnection fabric are registered, point-to-
2 point signals, enabling practitioners of the present invention to add an arbitrary number
3 of pipeline stages for timing or other purposes during design, layout, or modification of
4 the SOC.

Description Of Th Drawings

[07] To further aid in understanding the invention, the attached drawings help illustrate specific features of the invention and the following is a brief description of the attached drawings:

[08] FIG. 1 shows a standard computer workstation 10 of the type commonly used and suitable for SOC and other chip design activities.

[09] FIG. 2 shows a conceptual diagram of the present invention 100.

[10] FIG. 3 shows an example of a requestor connection port structure.

[11] FIG. 4 shows the structure of two types of target connection ports and the internal switching fabric included in the present invention.

[12] FIG. 5. is a block diagram of a typical decoder/router element 302.

[13] FIG. 6. shows an example four-requestor/three-target system that uses the present invention.

[14] FIG. 7 shows a second example system that uses the present invention, having five requestors and five targets.

Detail d D scription Of The Invention

[15] The present invention is a design framework and approach that enables SOC designers to develop flexibly upgradeable, complex, high-memory-bandwidth SOC designs that are capable of efficient verification and ready for the market in a reasonable amount of time. This disclosure describes numerous specific details that include specific structures, circuits, and logic functions in order to provide a thorough understanding of the present invention. One skilled in the art will appreciate that one may practice the present invention without these specific details.

[16] The Matrix Fabric framework of the present invention is used in system-on-chip designs containing one or more requestors for a shared system resource, which is typically, but not limited to, a memory device. In this description, a "requestor" is a functional module that makes a request to either read data or information from a target in the system or write data or information to a target in the system. To illustrate, one common requestor is a central processing unit (CPU) that requests data and information from one or more targets for instruction code fetches, cache line fills, and data processing. Other requestors include direct memory access (DMA) controllers that transfer blocks of data to and from system memory, and external I/O interface peripherals that transfer blocks of data from the I/O interface to and from system memory. Examples of external I/O interface peripherals include Universal Serial Bus (USB) host and device interfaces, Ethernet 10/100 or Gigabit interfaces, Peripheral Component Interconnect (PCI) interfaces, and Integrated Disk Electronics (IDE).

1 [17] A "target" is a functional module that provides one or more data ports or
2 addressable locations that can be read or written by an external requestor. Typical
3 targets in system-on-chips include embedded SRAM, external Flash, and external
4 dynamic RAM (synchronous or double-data rate). A target can also be a single access
5 device that controls several possible targets. This might include a centralized memory
6 controller that controls an external Flash and external SDRAM and which can process a
7 single request to one of its targets.

8 [18] Not all "targets" are memory devices. Peripheral devices and bus bridges can
9 also be targets in the context of this disclosure. Examples of these kinds of targets
10 might include a PCI controller acting as a bridge to a PCI memory device, an IDE Host
11 Controller serving as a bridge to an IDE Target device, or a digital-to-analog converter
12 generating an analog signal.

13 [19] In a typical system-on-chip configuration, different requestors all need access to
14 system resources, which is often system memory. Many system-on-chip designs use a
15 single memory target for a variety of reasons, including simplicity of design and cost. In
16 these designs, all memory requestors must arbitrate for the target memory. The target
17 system memory throughput is generally determined by the maximum throughput of the
18 target memory and the clock frequency of the target. For example, if the target memory
19 is a 32-bit wide internal SRAM that is accessible every clock cycle, the maximum
20 possible throughput for this system is 4 bytes per clock cycle. A system running at 100
21 MHz then would have a memory throughput of 400 Mbytes per second. In single target
22 systems, memory bandwidth can only be increased by expanding the throughput of the
23 target memory (e.g. using a 64-bit memory, or by increasing the clock frequency). In

1 this same single target system, using a 64-bit internal SRAM running at 100MHz would
2 increase the total throughput to 8 bytes per clock cycle, or 800Mbytes per second at
3 100MHz. Running this system at twice the clock speed would double this to 1.6Gbytes
4 per second.

5 [20] Ordinarily, requestors in a single target system will not require access to the
6 same region of memory at the same time. In the example of a single target memory
7 controller which supports separate Flash and SDRAM address spaces, one requestor
8 may want to read from the Flash while the other requestor may want to write the
9 SDRAM. Since there is only a single target, both requestors must arbitrate for
10 memory and one of them will have to wait until the other requestor completes its
11 transfer.

12 [21] Similarly, in some systems, certain address spaces are only accessible by
13 specific requestors. For example, in a multi-CPU system, processor instruction fetches
14 and cache line fills only occur from one address range in Flash space, while networking
15 packets from Ethernet interfaces are stored in a different SDRAM address range. In
16 these systems, even though there is no danger of two requestors trying to access the
17 same area of memory, both requestors must still arbitrate for access to the single
18 memory target.

19 [22] In both of these types of systems, if the architecture were redesigned such that
20 the different address spaces were separate targets, simultaneous and parallel access
21 could be allowed, thus increasing system throughput. In this approach, the second
22 target would exist in a different address range in system memory and could be

1 accessible by one or more of the memory requestors. Memory bandwidth is increased
2 when the different memory requestors do not all access the same memory target at the
3 same, with the peak memory throughput being the sum of the maximum bandwidths of
4 each of the individual targets. A multi-memory target system with an internal 32-bit
5 SRAM accessible every cycle and an external 64-bit SDRAM accessible every cycle will
6 have a peak bandwidth of 12 bytes per cycle (4 bytes per cycle from the 32-bit SRAM
7 and 8 bytes per cycle from the 64-bit SDRAM), or 1.2Gbytes per second when running
8 at 100MHz. Adding a third or even more memory targets is also possible, and would
9 increase overall system bandwidth accordingly when all targets are concurrently
10 accessible.

11 [23] The tradeoff designers face when adding extra targets is the increased system
12 design complexity. In most systems, adding another target means that each requestor
13 must now be modified to add in a new set of control and data signals to communicate
14 with the new target, and the SOC layout must be modified to add data paths between
15 the requestors and the new target. To illustrate, consider an example system with a
16 CPU and seven DMA memory requestors all accessing a single memory target. If a
17 second memory target is added, then all of the memory requestors must be modified to
18 add in the appropriate control and data path logic to communicate with this new target.
19 If, later in the design cycle, the architecture is enhanced to add a third memory target,
20 all of the requestors and the system design must be modified again. If the decision is
21 made on a multi-target system to revert back to a single target system with higher
22 throughput (e.g. switching from two 32-bit memory targets to a single 64-bit memory
23 target), then all of the designs must be changed again. Making these kinds of changes

1 during the design cycle always results in increased design and verification time, and
2 usually increases the overall complexity of the chip.

3 [24] The Matrix Fabric design framework was invented in order to solve these
4 problems. The framework supports a wide range of configurations, from a single
5 requestor and a single target to multiple requestors and multiple targets, rendering the
6 Matrix Fabric suitable for a variety of applications, from lower bandwidth and lower cost
7 designs to higher performance and higher bandwidth systems.

8 [25] The Matrix Fabric provides flexibility for adding requestors and targets to a
9 system-on-chip design, either during the initial design process or during subsequent
10 upgrades. In designs using the present invention, requestors do not need to know what
11 targets are available. Adding targets has no impact on the requestor design, and only
12 minimal changes are required to the Matrix Fabric itself. Adding requestors requires
13 adding an extra standard interface connection port to the Matrix Fabric; as each
14 requestor requires only a single interface connection port to the Matrix Fabric, as
15 described in greater detail below.

16 [26] The Matrix Fabric decodes all requests and routes them to the appropriate
17 target. Arbitration for the targets can be determined either by the target itself or by an
18 arbiter built into the Matrix.

19 [27] The Matrix Fabric takes a “building block” approach to interconnecting requestors
20 and targets, where the building blocks include standard requestor and target connection
21 ports, a decoder/router element per requestor, and an optional arbitration unit for each
22 target. Abstraction of the entire fabric into a single module allows for easier modification

1 and maintenance. When requestors and targets are to be added or removed, only one
2 functional module has to be updated rather than making changes across different
3 modules throughout the entire chip.

4 [28] The architecture of the Matrix Fabric allows for requestors and targets to be
5 easily added. Adding a requestor involves adding the requestor connection port and a
6 decoder/router element. Adding a target involves adding the target connection port and
7 updating the decoder/router element(s). Because the design is simple, these changes
8 can easily be made by hand. In addition, the regularity of the building block structures
9 of the Matrix Fabric make this interconnection architecture well suited for automatic
10 generation of register transfer level (RTL) code using computer scripts or other
11 software.

12 [29] The Matrix Fabric supports arbitrary pipelining, meaning that during the design or
13 physical layout of the system-on-chip, designers are free to add pipeline stages
14 between requestors and targets for timing or other purposes, without adversely
15 affecting the synchronization of the logic. All signals routed from the decoder/router
16 element(s) in the Matrix Fabric to either the optional arbiters or to the memory target
17 ports are point-to-point and registered, meaning that the signals are not directly
18 connected to functional logic at either their start or termination point, but instead, are
19 launched and captured by flip-flops. Thus, pipeline stages can be hidden inside the
20 Matrix Fabric structure. The bus protocols of the input and output ports are preferably
21 fully registered, so that pipeline stages can also be added to the input and output ports
22 of the Matrix Fabric. Arbitrary pipelining support helps solve the problem of timing
23 issues when the physical design of the chip grows larger, resulting in longer wiring

1 delays, or when the clock frequency increases. As a result, the fabric can be used in
2 both small and large designs, and in high-frequency and low-frequency designs.

3 [30] FIG. 1 shows a standard computer workstation 10 of the type commonly used
4 and suitable for SOC and other chip design activities. The computer workstation 10
5 shown in FIG. 1 is suitable for practicing the design and modification aspects of the
6 present invention discussed herein, and may also incorporate SOCs utilizing the
7 present invention. Those skilled in the art will understand that SOCs that incorporate
8 the present invention may also be used in any of a number of platforms, including but
9 not limited to handheld devices such as personal data assistants, communications
10 devices, servers, mainframes, embedded systems, laptops, and consumer electronics.

11 [31] As shown in FIG. 1, the workstation 10 comprises a monitor 20 and keyboard 22,
12 a processing unit 12, and various peripheral interface devices that might include
13 removable media local storage 14 and a mouse 16. Processing unit 12 further includes
14 internal memory 18, and internal storage (not shown in FIG. 1) such as a hard drive.

15 [32] Workstation 10 interfaces with digital control circuitry 24 and executable software
16 28 that may include, for example, device design and layout software if the computer
17 workstation 10 is functioning as a device design and layout workstation. In the preferred
18 embodiment shown in FIG. 1, digital control circuitry 24 is a general-purpose computer
19 including a central processing unit, RAM, and auxiliary memory. Both the executable
20 software 28 and the digital control circuitry 24 are shown in FIG. 1 as residing within
21 processing unit 12 of workstation 10, but both components could be located in whole or
22 in part elsewhere, and interface with workstation 10 over connection 26 or via

1 removable media local storage 14. As shown in FIG. 1, connection 26 could be a
2 connection to a network of computers or other workstations, which could also be
3 connected to printers, external storage, additional computing resources, and other
4 network peripherals. One skilled in the art will recognize that the software design and
5 layout aspects of the present invention can be practiced upon any of the well known
6 specific physical configurations of standalone or networked design workstations.

7 [33] The operator interfaces with digital control circuitry 24 and the software 28 via
8 the keyboard 22 and/or the mouse 16. Control circuitry 24 is capable of providing
9 output information to the monitor 20, the network interface 26, and a printer (not shown
10 in FIG. 1).

11 [34] FIG. 2 shows a conceptual diagram of the present invention 100. Conceptually,
12 the Matrix Fabric 100 can be broken into three sections: the connection ports to the
13 requestors 101, the connection ports to the targets 102, and the internal switching
14 fabric 103.

15 [35] As discussed in further detail below, each connection port includes standard
16 requestor control and data signals that would otherwise go to a generic target. These
17 signals should be part of a system-on-chip bus protocol and typically include, but are
18 not limited to, address, read/write direction, read/write data, and the appropriate control
19 signals. Any requestors can be connected to any connection port in the Matrix Fabric,
20 and there is no limit to the number of requestors that the present invention can
21 accommodate.

1 [36] Since each requestor is connected to the Matrix Fabric through a port, the
2 implementation of the connections results in a regular structure. The addition of another
3 requestor can be performed by copying an existing port module having the same
4 interface. As described above, the repetitive arrangement of the structure is highly
5 adaptable to the automatic generation of RTL code using computer scripts or other
6 software executing on a design workstation such as that shown in FIG. 1.

7 [37] FIG. 3 shows an example of a requestor connection port structure. FIG. 3
8 includes three requestors: requestor 0 201, requestor 1 202, and requestor X 203. As
9 shown in FIG. 3, in this example, each requestor connection port includes a standard
10 set of signals including a bus request signal (e.g., mb_init0_req); various data and
11 control strobes (e.g., mb_init0_astb, mb_init0_wstb, and mb_init0_rstb); a flow control
12 signal (e.g., mb_init0_rdy); a read/write control signal (e.g., mb_init0_dir); a target
13 address signal (e.g., mb_init0_addr); and data signals (e.g., mb_init0_rdata and
14 mb_init0_wdata). Adding a connection port for another requestor with the same
15 interface signaling requires only copying the requestor X signals and changing the X to
16 something else, e.g. requestor '2'. Those skilled in the art will understand that the
17 number, name, and types of specific signals included in each connection port may vary
18 as a matter of design choice, and the signal types, names, and number of signals
19 shown in FIG. 3 are not intended to convey any limitation of the present invention to the
20 signals shown.

21 [38] FIG. 4 shows the structure of the two types of target connection ports included in
22 the present invention. As shown in FIG. 4, a target with built-in arbitration 303 receives
23 a signal from each decoder/router channel 302 within the switching fabric 103. These

1 signals are routed to the target's arbitration port. Targets with no arbitration receive a
2 single set of signals from an arbiter 305 built into the switching fabric 103. The
3 switching fabric portion of the present invention, including the decoder/router channel
4 302 and the built-in arbiter 305, is described in further detail below.

5 [39] FIG. 4 also displays the structure of the internal switching fabric 103. The internal
6 switching fabric 103 includes one or more special decoder/router elements 302. Each
7 decoder/router unit 302 is connected to a single requestor through a requestor
8 connection port. The decoder/router unit 302 receives a request from its associated
9 requestor and routes this to the designated target using an internal system memory
10 map that contains the address ranges to which each target connected to the internal
11 switching fabric 103 via a target connection port is mapped. In a preferred embodiment,
12 the internal system memory map comprises a central memory map file included in the
13 decoder design. Each target is mapped to a pre-defined address range; the decoder
14 reads the address of the request and uses the internal system memory map to route
15 the request to the designated target(s).

16 [40] After reading this specification and/or practicing the present invention, those
17 skilled in the art will understand that the decoder/router unit design in the Matrix Fabric
18 enables the present invention to support different system-on-chip bus protocols. The
19 requestors can implement one system-on-chip bus protocol, while the targets can
20 support a different protocol. In addition, each requestor and each target may use the
21 same system-on-chip bus protocol or each may use any number of different system-on-
22 chip bus protocols. This feature allows more flexibility when integrating different design
23 components. As described in further detail below, the decoder/router elements translate

1 requests framed in the requestor bus protocol and route the requests to the appropriate
2 target(s) in the target system bus protocol.

3 [41] A block diagram of a typical decoder/router element 302 is detailed in FIG. 5.
4 The decoder/router 302 interfaces directly to the requestor connection port 101.
5 Requests are received by the request control flow block 403, which stores requests and
6 controls when requests are issued to the targets and when data transactions complete.
7 The address decoder block 404 decodes the incoming address of each request and
8 determines its intended target by using an internal system memory map 410 that
9 identifies which address spaces belong to each target. Once the target is determined,
10 the router logic 405 routes requests 412 to their designated target(s).

11 [42] The internal switching fabric provides flexibility regarding communication
12 between specific requestors and specific targets. Oftentimes, some requestors in a
13 multiple-requestor/multiple-target system do not need access to all of the targets. For
14 example, consider a four-requestor/two-target system comprising two CPUs and two
15 peripheral I/Os (the four requestors) and a flash controller and an SDRAM controller
16 (the two targets). In this example system, all four requestors require access to the
17 SDRAM but only the two CPUs require access to the flash. In this case, the internal
18 switching fabric can be set up so that all four requestors connect to the SDRAM but
19 only the two CPU's connect to the flash controller. This optimization saves logic, area
20 and routing congestion.

21 [43] To implement the above approach, individual decoder/router elements 302 are
22 designed for each combination of targets that a requestor requires. For example, if a

1 requestor requires access to only a single target, a single target decoder/router element
2 is created which has only one request output port. If a memory requestor requires
3 connections to three different targets, then the decoder/router element uses three
4 different request output ports.

5 [44] In many systems, all of the requestors are allowed access to all of the targets,
6 and thus the same design of a decoder/router element 302 can be used for all
7 requestor ports. This allows for simplicity in adding new requestors and targets. When a
8 new requestor is added, the internal switching fabric 103 requires only an additional
9 decoder/router element 302. If a new target is added, the existing decoder/router
10 element(s) need(s) a new memory target port. These design changes to the source
11 design descriptions can easily be performed by hand, or automatically through use of
12 computer scripts or other software executing on a workstation such as that shown in
13 FIG. 1.

14 [45] Systems may have two or more different types of decoder/router elements in the
15 internal switching fabric. For example, systems wherein some requestors do not
16 require access to all targets may have a two-target decoder and a three-target decoder
17 to handle the different requestor/target paths. However, typically only a few different
18 types of decoder/routers are ever required in most system implementations. Because of
19 the regular structure of the Matrix Fabric, at most only a few decoder/router elements
20 need to be designed; combinations of the decoder/router elements can create all of the
21 desired designs. Alternatively, computer scripts or other software executing on a
22 workstation can be used to automatically generate any required combination of
23 decoder/router element designs.

1 [46] An example system 500 that uses the Matrix Fabric of the present invention is
2 shown in FIG. 6. In this system 500 there are four requestors 501 (CPU1 507, CPU2
3 508, and two DMA peripherals 509 and 510) and three targets 515 (a controller for
4 external flash 503 used for code execution, a controller for external SDRAM 504 used
5 for main system memory, and a controller for high speed internal SRAM 505). Each
6 requestor is connected to a decoder/router element (502, 511, and 512) in the internal
7 switching fabric 550 via a requestor connection port 520. Each target is connected to
8 the internal switching fabric 550 via a target connection port 540. The decoder/router
9 elements receive the input request and map these to the appropriate target based on
10 the address of the request.

11 [47] Example system 500 illustrates several of the features of the present invention.
12 The first target, the external flash controller 503, is a slave that has no internal
13 arbitration, so an arbitration unit 506 for this target is built into the switching fabric 550.
14 In addition, since the only requestors that require access to the external flash 503 are
15 the two CPUs 507 and 508, these are the only requestors connected to this target via
16 router/decoder elements.

17 [48] The second and third targets are an SDRAM memory controller 504 and an on-
18 chip SRAM controller 505, respectively. Both of these targets are accessible by all of
19 the requestors, and both targets also have internal arbitration. Accordingly, since the
20 two CPUs require access to all three targets, but the two DMA peripherals require
21 access to only two of the targets, the CPUs each use a “three-target” decoder/router
22 element 502, while the two DMA requestors each use a “two-target” decoder/router
23 element 511, 512.

1 [49] FIG. 7 shows a second example system 600 that uses the Matrix Fabric of the
2 present invention. System 600 has five requestors and five targets. The five requestors
3 include a CPU 601, a DMA controller 602, an Ethernet 10/100 peripheral 603, a USB
4 2.0 Host peripheral 604, and the master interface 605 of a PCI bridge. The targets
5 include a single port memory controller 606 that controls a separate external flash and
6 separate SDRAM controller, a dual port internal SRAM 607 having separate read and
7 write ports, a IDE Host Controller 608, and the slave interface 609 of the PCI Bridge
8 listed above. All requestors connect to the switching fabric 610 via requestor
9 connection ports 620. All targets connect to the switching fabric 610 via target
10 connection ports 630.

11 [50] The FIG. 7 example system illustrates some aspects of the present invention not
12 covered in the FIG. 6 system. In system 600, the same PCI Bridge functions both as a
13 requestor 605 and a target 609. The PCI Bridge contains a master interface 605 that
14 generates requests to other targets in the system 600. The PCI bridge also has a
15 separate target interface 609 that allows the bridge to receive and process requests
16 from the other requestors in the system. In this example, the PCI Bridge master 605
17 can generate requests that are routed through the internal switching fabric 610 destined
18 for the IDE Host Controller 608 and the shared flash/SDRAM controller 606. The PCI
19 Bridge slave 609 can receive requests that have been routed through the internal
20 switching fabric 610 from the CPU 601, the Ethernet peripheral 603, and the USB host
21 604. The structure of the Matrix Fabric allows a single device—in this case a PCI
22 bridge—having two separate ports to act as both a requestor and a target.

1 [51] Similarly, the Dual-Port internal SRAM controller 606 is a single device that acts
2 as two separate targets, since each port can be independently accessed. As shown in
3 FIG. 7, each port has its own built-in arbiter. Therefore, in system 600, reads from the
4 SRAM can occur simultaneous with writes to the SRAM.

5 [52] The IDE Host Controller target 608 and the PCI Controller target 609 both act as
6 bridges to other devices/systems. Both of these device bridges are designed as
7 targets, having a target interface, so that they are addressable by a requestor. This
8 design approach allows transfers to occur from the Ethernet device 603 or USB 2.0
9 device 604 through the switching fabric 610 directly to the IDE Host Controller 608 or
10 the PCI Controller 609.

11 [53] In summary, the present invention is a System-on-Chip (SOC) interconnection
12 apparatus and system, wherein an internal switching fabric interconnects one or more
13 requestors and one or more targets on a single semiconductor integrated circuit. Each
14 target has a unique address space, may or may not have its own arbitration, and may
15 be resident (i.e., on-chip) memory, a memory controller for resident or off-chip memory,
16 an addressable bridge to a device, system, or subsystem, or any combination thereof.
17 Targets and requestors are connected to the internal switching fabric of the present
18 invention using target connection ports and requestor connection ports.

19 [54] Signals are routed between requestors and targets using one or more
20 decoder/router elements within the internal switching fabric. Each decoder/router
21 element receives a request from a requestor, determines which target is the designated
22 target using an internal system memory map, and routes the request to the designated

1 target. The internal system memory map used in an individual decoder/router element
2 may include unique address space information for all of the targets in a system, or
3 fewer than all of the targets in a system. A single decoder/router element may route
4 requests to all of the targets in a system, or fewer than all of the targets in a system.

5 [55] The internal switching fabric may also include independent memory arbiters
6 dedicated to memory targets that do not have internal arbitration. Finally, the signals
7 routed between the decoder/routers and the memory targets by the interconnection
8 fabric are registered, point-to-point signals, enabling practitioners of the present
9 invention to add an arbitrary number of pipeline stages for timing or other purposes
10 during design, layout, or modification of the SOC.

11 [56] Other embodiments of the invention will be apparent to those skilled in the art
12 after considering this specification or practicing the disclosed invention. The
13 specification and examples above are exemplary only, with the true scope of the
14 invention being indicated by the following claims.